IN-61
43106
P.8

Human Error and the Search for Blame

Peter J. Denning

29 Nov 89

RIACS Technical Report TR-89.46

NASA Cooperative Agreement Number NCC 2-387

Research Institute for Advanced Computer Science
An Institute of the Universities Space Research Association

Human Error and the Search for Blame

Peter J. Denning

Research Institute for Advanced Computer Science
NASA Ames Research Center

RIACS Technical Report TR-89.46
29 Nov 89

Human error is a frequent topic in discussions about risks in using computer systems. A rational analysis of human error leads us through consideration of mistakes to standards that designers use to avoid mistakes that lead to known breakdowns. The irrational side, however, is more interesting. It conditions us to think that breakdowns are inherently wrong and that there is ultimately someone who is responsible. This leads on on a search for blame that distracts us from learning from our mistakes, from seeing the limitations of current engineering metholodgy, and from improving our discourse of design.

This is a preprint of and editorial for
Communications of ACM 33 78, No. 1 (January 1990).

Work reported herein was supported in part by Cooperative Agreement NCC 2-387
between the National Aeronautics and Space Administration (NASA)
and the Universities Space Research Association (USRA).

Human Error and the Search for Blame

Peter J. Denning

Research Institute for Advanced Computer Science

29 Nov 89

The ACM RISKS Forum, moderated by Peter Neumann of SRI International, is an electronic dialog among a community of network users concerned about risks to the public in computer systems. A frequent topic is the role of human error in mishaps involving computers. These discussions have raised interesting questions for me. What is the phenomenon we call human error? Why does it keep coming up so often in discussions about risks of using computers?

Most of us associate errors with mistakes. When I say I made a mistake, I usually mean one of two things. I may mean that I misjudged the consequences of an action I took and the consequences had an unwarranted or unreasonable cost that must now be compensated. Or I may mean that I had no way of foreseeing the unintended consequence, and now I regret having taken the action. In the first case I had a choice but took action in the face of the risk, while in the second case I was blind to the consequences and had no real choice at all.

But these notions about mistakes are rational. In the background of our thinking about mistakes is an automatic, irrational reaction that they are wrong. Our emotions have not been conditioned to regard mistakes as basic constituents of the human condition or as opportunities to learn; our emotions have us think instead that mistakes shouldn't have happened and that their occurrences are problems in need of correction. I will return to this point shortly. Let me continue with the rational side.

When certain actions recurrently have unwanted consequences, we usually establish standards that tell us what action is desired or what action is to be avoided. Such standards evolve in a community over time. When we are about to take an action we can evaluate it against the standard; we do not have to wait until some time later when the unwanted consequence appears.

The domain of engineering practice illustrates this. The designer of a communication device is expected to use accepted error-correcting codes to resist transmission noise. That designer commits a design error by omitting the codes; we say he may be held liable if a user of the equipment encounters a mishap because an erroneous signal was accepted by the device. The important point here is that there are accepted practices and that the engineer in question did not follow them.

But what about the case where there are no standards or accepted practices? The designer has no standard against which to assess the design. The community has no way to assess whether a design is risky -- it is a new case. Software engineering is like this. Having noted the recurrent difficulties of constructing dependable software systems, some observers of the field of software engineering have begun to suspect that the current practices of the field (including methodologies and theories) are incapable of producing

dependable software -- something unseen is missing. A designer operating within these currently accepted practices is therefore likely to produce systems that produce breakdowns that he could not foresee. In what sense is he responsible for the error?

How we answer this depends on how we understand responsibility. Our ordinary everyday understanding associates responsibility with fault, blame, guilt, and liability. We have been trained by our legal system to think in terms of evidence to prove who is at fault and of compensation to the injured parties. The search for blame digs deeply. We live our lives under the supposition that somewhere, there is someone or some organization that is ultimately responsible for everything. Our litigiousness keeps us constantly looking for something to blame and someone to hold liable. We persist in this search despite the evidence of everyday observation that many human practices "just evolved" without anyone's planning them or anticipating them.

Look at how this way of thinking inclines us to interpret breakdowns produced by software systems. It is always easy, with 20/20 hindsight, to say that the design that has produced a breakdown was weak or flawed. We are automatically drawn into a search for blame. Our first target is the designer. Are we justified in saying that the designer, who operated without the benefit of the hindsight, can be held liable for the flaw? Some of us say that it is unfair to blame the designer who operated in good faith within the standards of the field; we look elsewhere for the blame. Our next target may be the organization that employs the designer. If we continue the search long enough we will be led to saying that the designers of the standard practices are to blame. This can bring us to the curious position that the authors of software engineering textbooks, or the officers of the professional societies that recommended practices and curricula, are to

blame for flaws detected in systems designed with these methodologies. This sounds almost preposterous -- except that I have read accounts of exactly such proposals!

Let us consider another interpretation of responsibility, as it is used in phrases such as "professional responsibility." Here responsibility is a declaration that a person or organization has influence on the outcome or on people's interpretation of the outcome, and is prepared to take action consistent with that declaration. This understanding of responsibility would have us act as follows.

1. We would honor the standards the community has established in the domain in which we take responsibility;
2. Having observed recurrent breakdowns, we would work to alter the standards and practices of the domain to anticipate the breakdowns in the future;
3. We would be prepared to admit and learn from our mistakes and to extend the same privilege to others; and
4. As a community, we would be open to being shown systematic blindnesses in our standard practices and to taking actions to improve them.

Within this interpretation, a responsible software designer would make it a point to know and abide by the standards and practices of software engineering; and the designer would forever be a student of the breakdowns of software systems and would where necessary work to alter the standards of the field. New practices such as rapid prototyping, design teams including users, participatory design and anonymous databases of design flaws and mishaps can all contribute to overcoming our blind spots and to supporting us in these responsibilities.

The background supposition that there is always someone to blame is an illusion blinds us to effective action within the domain of engineering. The search for blame prevents us from seeing the limitations of current engineering methodology and distracts us from improving our discourse of design. Indeed, the fear of being singled out for blame may well disincline us from trying courses of action whose consequences cannot be foreseen -- the fear of our own blindness perpetuates our blindness.